

1. 如何阅读本章

系统滴答定时器 (SysTick 定时器)是 ARM Cortex-M0 核的一部分, 所有LPC111x系列处理器的系统滴答定时器均相同。

2. 特性

- 简单24位定时器。
- 使用专用的异常向量。
- 由一个专门的内部系统时钟定时器时钟提供时钟, 该时钟也是ARM核时钟。

3. 描述

SysTick 定时器是Cortex-M0内部集成的一部分。SysTick定时器的作用是产生一个固定的10毫秒中断, 供操作系统或其它系统管理软件使用。

由于SysTick定时器是Cortex - M0的一部分, 它为基于Cortex - M0的处理器提供一个标准计时器, 有助于软件的移植。SysTick定时器可用于:

- 可编程设置频率的RTOS定时器 (例如100 Hz), 调用一个SysTick服务程序。
- 用于核时钟的高速报警定时器。
- 简单计数器。软件可使用它测量时间(如: 完成任务所需时间、已使用的时间)。
- 基于丢失/命中期限控制的内部时钟源。控制和状态寄存器中的COUNTFLAG位域, 可用于决定一个动作是否在设定的期限内完成, 作为动态时钟管理控制环的一部分。

详情可参考 *Cortex - M0* 用户指南。

4. 操作

SysTick定时器是一个24位定时器, 向下计数到零, 就会产生一个中断。这样可以提供一个固定的10毫秒间隔的中断。SysTick定时器的时钟来自CPU时钟。为了连续产生特定时间间隔的中断, SYST_RVR寄存器必须用正确的值进行初始化, 以保证获得所需的时间间隔。SYST_CALIB寄存器中有一个默认值, 可以通过软件修改; 如果设置为默认值, 则会每10毫秒产生一次中断。

SysTick定时器的框图如图 15-49所示。

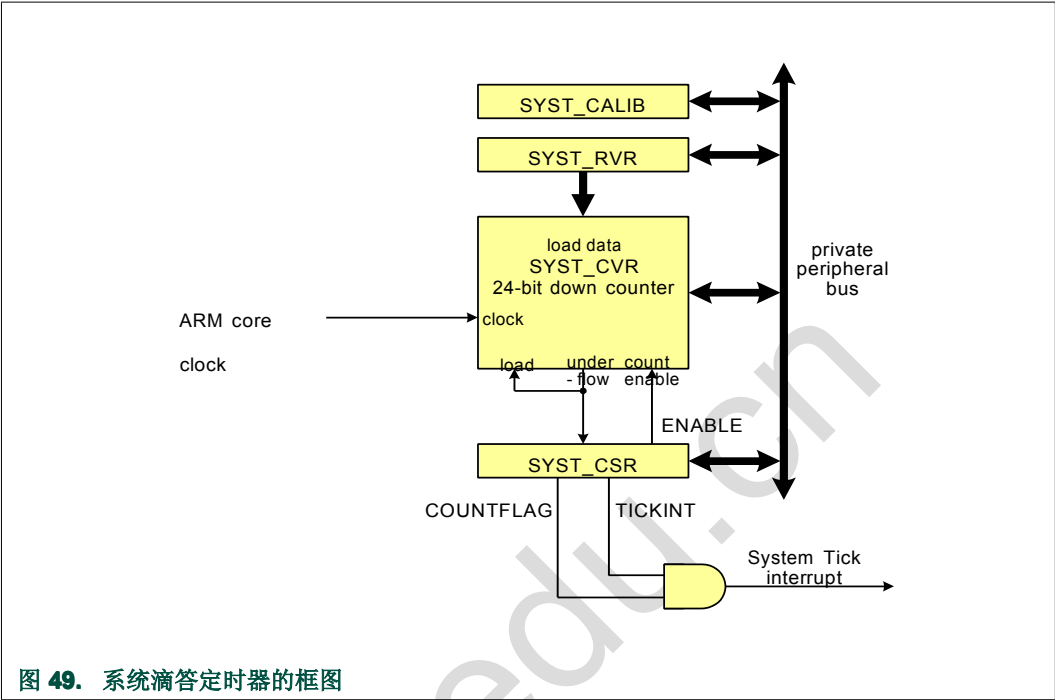


图 49. 系统滴答定时器的框图

5. 寄存器描述

Systick定时器位于ARM Cortex-M0 专用外设总线上(见图19-55), 是ARM Cortex-M0核内外设。详见19-5.4小节。

表 202. SysTick 定时器 寄存器概览(基地址 0xE000 E000)

名称	访问方式	地址偏移	描述	复位值 ^[1]
SYST_CSR	R/W	0x010	系统定时器控制和状态寄存器	0x000 0000
SYST_RVR	R/W	0x014	系统定时器重载寄存器	0
SYST_CVR	R/W	0x018	系统定时器当前计数值寄存器	0
SYST_CAL	R/W	0x01C	系统定时器校准值寄存器	0x4

[1] 复位值只反映使用位的值，不包括保留位的内容。

5.1 系统定时器控制和状态寄存器

SYST_CSR寄存器包含SysTick 定时器的控制信息，并提供了一个状态标志。该寄存器是ARM Cortex-M0内核系统定时器寄存器模块的一部分。该寄存器的位域描述，见19-5.4.1小节“SysTick 控制与状态寄存器”。

5.2 系统定时器重载寄存器

每当SysTick定时器计数减少到零时，会将SYST_RVR寄存器中设置的值装载到SysTick定时器中。软件加载该寄存器可以作为定时器初始化的一部分。如果CPU运行在为了使用SYST_CAL值的频率时，则SYST_CAL寄存器可读取并作为STRELOAD的值。

表 203. 系统定时器重载寄存器 (SYST_RVR - 0xE000 E014) 位域描述

位	符号	描述	复位值
23:0	RELOAD	每当SysTick定时器计数减少到零时， 将要到装载这个值到SysTick定时器。	0
31:24	-	保留。用户软件不应该对保留位写 ‘1’ 。从保留位读取的值没有定义。	NA

5.3 系统定时器当前值寄存器

当软件读SYST_CVR寄存器时，返回系统滴答计数器当前的计数值。

表 204. 系统定时器当前值寄存器 (SYST_CVR - 0xE000 E018) 位域描述

位	符号	描述	复位值
23:0	CURRENT	读该寄存器，将返回系统滴答计数器当前的计数值 对该寄存器写任何值都会清除系统滴答计数器和STCTRL寄存器的COUNTFLAG位。	0
31:24	-	保留。用户软件不应该对保留位写 ‘1’ 。从保留位读取的值没有定义。	NA

5.4 系统定时器校准值寄存器 (SYST_CAL - 0xE000 E01C)

<待定义>

表 205. 系统定时器校准值寄存器 (SYST_CAL - 0xE000 E01C) 位域描述

位	符号	描述	复位值
23:0	TENMS	<待定>	<待定>
29:24	-	保留。用户软件不应该对保留位写 ‘1’ 。从保留位读取的值没定义。	NA
30	SKEW	<待定>	0
31	NOREF	<待定>	0

6. 定时器计算举例

按如下步骤使用系统滴答定时器:

1. 用重载值RELOAD对SYST_RVR寄存器进行编程设置, 以获得要求的时间间隔;
2. 通过写SYST_CVR寄存器来清除该寄存器。这将确保定时器被允许时, 定时器将从SYST_RVR的值开始计数, 而不是从任意值开始计数。
3. 用值0x7设置寄存器SYST_SCR, 以允许SysTick定时器和SysTick定时器中断。

下面的例子说明了不同系统配置时SysTick定时器的重载计数值。所有例子计算的中断间隔都为10毫秒, 因为SysTick定时器通常如此使用, 而且无舍入错误。

例 (system clock = 50 MHz)

$\text{RELOAD} = (\text{system clock frequency} \times 10 \text{ ms}) - 1 = (50 \text{ MHz} \times 10 \text{ ms}) - 1 = 500000 - 1 = 499999 = 0x0007A11F$

例 (system clock = 40 MHz)

$\text{RELOAD} = (\text{system clock frequency} \times 10 \text{ ms}) - 1 = (40 \text{ MHz} \times 10 \text{ ms}) - 1 = 400000 - 1 = 399999 = 0x00061A7F$

例 (system clock = 12 MHz)

这种情况下, 系统时钟从IRC时钟获得。

$\text{RELOAD} = (\text{system clock frequency} \times 10 \text{ ms}) - 1 = (12 \text{ MHz} \times 10 \text{ ms}) - 1 = 120000 - 1 = 119999 = 0x0001D4BF$